Few-Shot Self Reminder to Overcome Catastrophic Forgetting

Junfeng Wen[†, ‡, *] **Yanshuai Cao**[†] **Ruitong Huang**[†]
[†]Borealis AI [‡]University of Alberta
junfengwen@gmail.com {yanshuai.cao,ruitong.huang}@borealisai.com

Abstract

Deep neural networks are known to suffer the catastrophic forgetting problem. In this work, we present a simple yet surprisingly effective way of preventing catastrophic forgetting. Our method, called Few-shot Self Reminder (FSR), regularizes the neural net from changing its learned behavior by performing logit matching on selected samples kept in episodic memory from the old tasks. Surprisingly, this simplistic approach only requires to retrain a very small amount of data in order to outperform previous methods in knowledge retention. We demonstrate the superiority of our method to the previous ones on popular benchmarks as well as a new continual learning problem where tasks are designed to be more dissimilar.

1 Introduction

Neural networks suffer from catastrophic forgetting, a phenomenon in sequential learning of multiple tasks whereby previous knowledge is lost by mistake when new tasks are learned [12, 11, 3]. When the tasks are learned sequentially, optimization in later stage could adapt the shared parameters and representations in ways that harm the old tasks. This failure hampers the application of deep models since it indicates that they are incapable of maintaining knowledge when facing new environments.

Many different ways to address this problem have been explored in the literature. While the methods that store all the previous data face the resource challenge in practice, alternative methods that instead store models have been proposed in the literature [6, 17, 10, 8, 16, 4]. For example, Elastic Weight Consolidation (EWC) [6] stores the previous model to penalizes the model parameter changes according to the different sensitivities. Shin et al. [15] replaces the storage of the previous data by training GANs to generate fake historical data. Besides the complications of these methods, which usually leads to an exhausted hyper-paramter tuning, such approaches also need to pay significant cost for storing a reasonably 'good' model, given the size of the current state-of-the-art networks. Thus, such methods may not necessarily save its storage cost, especially when saving a small subset of "anchor" points from the historical data is sufficient for this problem, as we will show later.

In this work, we embrace simplicity and show that it is possible to address catastrophic forgetting by storing and reusing very few previous data without incurring significant memory cost. We propose Few-shot Self Reminder (FSR), which directly places the regularization on the function mapping instead of its parameters. It does so with a very small episodic memory of previous data and their corresponding logits. This idea is adopted from the model compression community [2, 1, 5], but used in a very different manner. FSR is frustratingly simple, but surprisingly effective in practice.

1.1 Related Work

The most related methods to FSR in the literature are Learning without Forgetting (LwF) [9] and iCaRL [14], which also use distillation to prevent catastrophic forgetting. In particular, LwF matches the predicted labels of previous models on the *current* data, which requires to store all the previous

[*] Work done while interning at Borealis AI

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

models. Its performance also drop sharply when the input distribution changes significantly across tasks. iCaRL focuses on class-incremental learning and matches independent logistics in the outputs for representation learning, which differs from their classifier objective. The different logits of independent logistics are not jointly calibrated, so they do not contain as rich information as logits of softmax classifiers, and behaves qualitatively different from FSR in practice.

2 Few-Shot Self Reminder

We focus on the continual learning setting, in which the learner will encounter a sequence of datasets $\mathcal{D}_1, \mathcal{D}_2, \cdots$, one at a time. The goal is to attain a model $f_T : \mathcal{X} \mapsto \mathcal{Y}$ that performs well on the first T datasets after sequentially trained on them, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the probability simplex. We assume the value of T is not known in advance so we would like to have a good model f_T for any T during the sequential training. This learning problem is challenging in that, if f_T is simply trained on the current dataset $\mathcal{D}_T$, it will forget how to properly predict for datasets $\mathcal{D}_t, t < T$, the so-called catastrophic forgetting problem. Denote the total loss by $L^{(T)}(f) = \sum_{t=1}^{T} \mathbb{E}_{\mathcal{D}_t} \left[L\left(f(X), Y\right)\right]$, where (X, Y) is the random data pair in dataset D_t and let $f_T \overset{\text{def}}{=} \operatorname{argmin}_f L^{(T)}(f)$. For simplicity, we denote $\mathbb{E}_{\mathcal{D}_t} \left[L\left(f(X), Y\right)\right]$ by $L_{\mathcal{D}_t}(f)$. The constraint is that we won't be able to access all the data from the previous tasks, but a limited amount of information stored in episodic memory.

Unlike EWC constraining the model parameters of f_T, our method FSR focuses on its function mapping. When learning a new task, our method carries over a small amount of information to "remind" the learner about the knowledge from the previous tasks. Noting that

$$\min_f L^{(T)}(f) = \sum_{t=1}^{T-1} L_{\mathcal{D}_t}(f_t) + \min_f \left[L_{\mathcal{D}_T}(f) + \sum_{t=1}^{T-1} \Delta_{\mathcal{D}_t}(f, f_t) \right],$$

where $\Delta_{\mathcal{D}_t}(f, f_t) = L_{\mathcal{D}_t}(f) - L_{\mathcal{D}_t}(f_t)$ measures the difference in the performances of f and f_t on $\mathcal{D}_t$. Therefore, learning f_T requires minimizing $L_{\mathcal{D}_T}(f) + \sum_{t=1}^{T-1} \Delta_{\mathcal{D}_t}(f, f_t)$.

It remains to decide what information from $\mathcal{D}_t$ is important for f to achieve a small $\Delta_{\mathcal{D}_t}(f, f_t)$, given a limited amount of memory. One approach would be to pass a small number of samples $\widetilde{\mathcal{D}}_t = \{(x_j^{(t)}, y_j^{(t)}) \mid j = 1, \cdots, m\}$ from $\mathcal{D}_t$ (or the predicted labels), thus $\sum_t \Delta_{\mathcal{D}_t}(f, f_t)$ can be replaced by $\sum_t \Delta_{\widetilde{\mathcal{D}}_t}(f, f_t)$. However, such approach depends heavily on that a single label can represent the structured output of f_t, which is unrealistic in general. In order to pass the information of the structured output to fully reproduce the predicting behavior of a model, we propose to the following "self-distillation" method on the logits:

$$\min_\theta \frac{1}{n_T} \sum_{i=1}^{n_T} L(f(x_i^{(T)}), y_i^{(T)}) + \frac{\lambda}{m} \sum_{t=1}^{T-1} \sum_{j=1}^{m} \|z_j - z_j^{(t)}\|_2^2,$$

where $z_j, z_j^{(t)}$ are the logits of the memory data x_j produced by f and f_t respectively.

Intuitively, the selected points $\widetilde{\mathcal{D}}_t$ should be representative and provide as much constraint as possible to change in f. Surprisingly, it turns out that class-stratified random sampling already works exceptionally well in our experiments, and other more sophisticated methods do not consistent outperform it with a significant margin. We also test out an efficient parameter-gradient based estimation method. The intuition is that representative points are both easier to learn (comparing to "corner cases") and occur more frequently in the training set. Hence as the initial transient phase of learning epochs, representative points should contribute less model parameter gradient on average. At later iterations, the norm of the gradients is equally assigned to all the points in the batch as their additive scores. After training one task, the points with the lowest scores are selected. Empirically, this method outperforms stratified random sampling, but not always with a significant margin.

3 Experiments

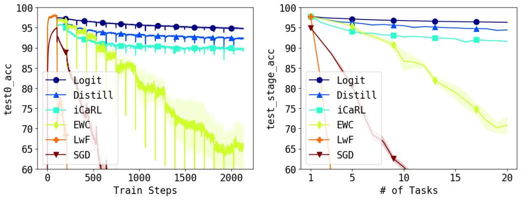
In this section, we empirically demonstrate that our proposed approach forgets much slower than popular alternatives and that it can handle highly dissimilar tasks. As FSR performs logit matching, it is labeled as *Logit*. We also test a variant with knowledge distillation [5] (labeled as *Distill*) using cross-entropy on softmax probabilities. The baselines are vanilla SGD, LwF [9], iCaRL [14] and EWC [6]. SGD is a naïve baseline used to showcase performance of EWC in Kirkpatrick et al. [6]

so we include it as comparison. LwF and iCaRL share some similarity to our method, but have important distinctions as outlined in previous sections. The regularization parameter of each method is individually tuned with a large range of candidates, based on a hold-out validation partition. Except SGD, all other methods are trained using the Adam with step size of 0.0001. We test our method on two different benchmark problems designed based on MNIST. Each setting has 20 tasks in total.

Permuted MNIST The first setting is the permuted MNIST problem [7], a popular benchmark for continual learning [6, 17, 10]. For each task, a fixed random permutation is applied to all inputs. As pixel permutation is a linear transformation, the resulting tasks are relatively similar to each other.

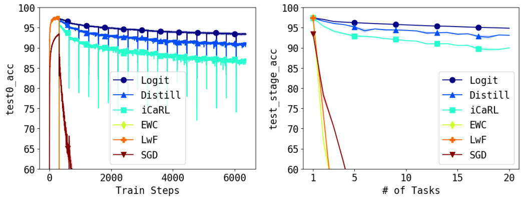
Nonlinearly Transformed MNIST To compare how the methods handle more dissimilar tasks, we further design a nonlinearly transformed MNIST benchmark. In this problem, a fixed random nonlinear (but invertible) transformation (a four-layer MLP with random orthogonal weights) is applied to all the images. All layers have 784 units with LeakyReLU ($\alpha = 0.2$) activation. Each task corresponds to a different random nonlinear transformation. The invertible transformations lose no information, so each task is still equally solvable by a permutation invariant model like MLP.

Our FSR can trade-off between the memory usage and knowledge retention. In the rest of the paper we will first demonstrate how FSR can forget slower than existing methods when using comparable memory, followed by several surprising results when applying FSR with very small memories.



(a) First Task (b) Task Average

Figure 1: Permuted MNIST test accuracy



(a) First Task (b) Task Average

Figure 2: Non-linear MNIST test accuracy

3.1 Little Forgetting

Permuted MNIST We randomly select 500 class-balanced MNIST images per task as memory for FSR, which in total is comparable to the memory cost of EWC on the same model.

Figure 1a shows the test accuracy of the first task along with the training of 20 sequential tasks, while Figure 1b shows the average test accuracy of tasks thus far. We can see that all methods except LwF outperform SGD with a large margin. LwF performs poorly in this problem due to two possible factors: (1) noticeable distribution changes in the input space, as also pointed out by other researchers [13] and (2) the fact that the two losses based on ground truth label and distilled label from previous task are in fact conflict with each other. It is difficult for a single model to predict both labels given the same image. Another observation from Fig. 1 is that, logit matching, distillation, and iCaRL have a significant improvement over EWC when using comparable memory size.

Nonlinearly Transformed MNIST We then test on the a more challenging problem of nonlinearly transformed MNIST, where tasks are less similar. The results are shown in Figure 2. As we anticipated, when data distributions are much different from task to task, approaches that match model parameters like EWC can fail miserably. Essentially, EWC only utilizes local information as the diagonal Fisher matrix. When the optimal solutions of two tasks are far apart, the local information of the first task is no longer accurate during the training process of the second task, and there might not be overlap for the two estimated Gaussian ellipsoids. On the contrary, methods that solely match the outputs of previous models like FSR can still maintain a remarkably better performance than EWC.

3.2 Little Memory

To further examine the effectiveness of FSR, we test our method with small memory. FSR can surprisingly do well in the extreme setting of retaining only a few images. We focus on the permuted MNIST setting and show the effect of different memory sizes in Figure 3. There are a few observations.

First, Adam-optimized models tend to forget more quickly than those optimized by vanilla SGD. It may be explained by the fact that adaptive optimizers usually find local optimum of the new task quicker than SGD, hence moving away from previous solutions more quickly. Second, strikingly, even with only 1 image per class (a memory size of 10 images per task), logit matching can improved over SGD by a significant margin. Recall that we match logits with the Adam optimizer, which means that even with only 1 image per class can remedy the forgetting issue of Adam. Third, with 10 images per class (thus 100 images per task), logit matching can outperform EWC for this problem. It is surprising that logit matching can perform so well with only 20% of the memory cost of EWC. Fourth, as shown in Figure 3, logit matching consistently performs better than distillation and iCaRL, across all memory sizes. Their accuracy differences are more significant with smaller memories.

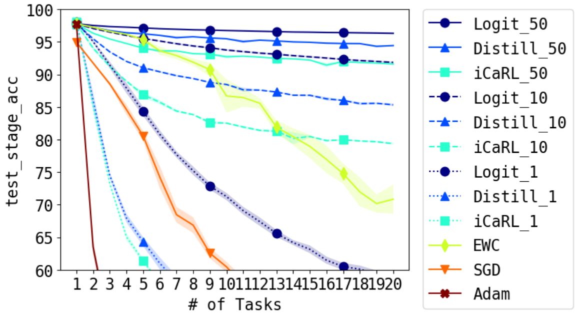


Figure 3: Permuted MNIST average task test accuracy with different memory sizes (mean and standard error over 5 repetitions). Solid lines are using full memory. Dashed and dotted lines are with partial memory. The numbers in the legend indicate the numbers of examples per class per task[2].

4 Conclusion

To overcome the catastrophic forgetting problem in continual learning of deep neural networks, we proposed Few-shot Self Reminder (FSR) that requires substantially smaller memory yet forgets slower than popular alternatives. As a side contribution, we also introduced a new benchmark, nonlinearly transformed MNIST, that is significantly harder with more substantial between-task dissimilarities.

References

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[2] Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006.

[3] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[4] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. *International Conference on Learning Representations*, 2018.

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[6] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662, 2017.

[9] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[10] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[11] James L McClelland, Bruce L McNaughton, and Randall C O'reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3): 419, 1995.

[12] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[13] Amal Rannen Ep Triki, Rahaf Aljundi, Matthew Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings ICCV 2017*, pages 1320–1328, 2017.

[14] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5533–5542. IEEE, 2017.

[15] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[16] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

[17] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995, 2017.

[2] For example, "Logit_1" means FSR with logit matching with only 1 example per class per task so each task will use a memory of size 10 images.